CLAIMS

What is claimed is:

1. A method comprising:

maintaining a first form of an intermediate result of an operation in a first register;

maintaining a second form of the intermediate result in a second register;

responsive to receiving digits *1* to *L-2* of the intermediate result from a digit recurrence unit, where *L* represents a number of digits that satisfies a predetermined precision for the operation, updating each of the first form and the second form of the intermediate result by register swapping or concatenation under the control of load and shift control logic and on-the-fly conversion logic;

generating a rounded result by determining digits d_{L-1} and d_L and deriving from these the two digits d'_{L-1} and d_L^{rnd} which are then appended to either the first form of the intermediate result or the second form of the intermediate result.

2. The method of claim 1, wherein the first form of the intermediate result comprises a value R_j representing a sum of intermediate digits, $d_1 r^{-1} + d_2 r^{-2} + ... + d_j r^{-j}$, which converges to an infinitely precise result as *j* tends to infinity, where *r* represents a radix associated with a digit-recurrence algorithm implemented by the digit-recurrence unit, *j* represents a current iteration of the digit recurrence algorithm, and d_j represents a digit generated during iteration *j* of the digit-recurrence algorithm.

3. The method of claim 2, wherein the second form of the intermediate result comprises a value R_j^-, where R_j^- differs from R_j by a unit of the j^{th} position, r^{-j}.

4. The method of claim 3, wherein said updating each of the first form and the second form of the intermediate result by register swapping or concatenation comprises updating R_j and R_j^- based upon d_j, *r*, and R_{j-1} and R_{j-1}^-.

5. The method of claim 1, wherein storage typically allocated to a value, R_j^+, representing $R_j + r^{-j}$ in conventional digit-recurrence procedures employing on-the-fly rounding is saved by not relying upon and not maintaining R_j^+ during iterations *1* to *L-2* of the digit-recurrence procedure.

6. The method of claim 1, wherein:

$$R_j = \begin{cases} (R_{j-1}, d_j) & d_j \geq 0 \\ (R_{j-1}^-, d_j + r) & d_j \leq -1 \end{cases}; \text{ and}$$

$$R_j^- = \begin{cases} (R_{j-1}, d_j - 1) & d_j \geq 1 \\ (R_{j-1}^-, d_j + r - 1) & d_j \leq 0 \end{cases}.$$

7. The method of claim 1, wherein said generating a rounded result further comprises:

generating the rounded last digit d_L^{rnd};

if $d_L^{rnd} \leq -1$, then modifying d_L^{rnd} and d_{L-1} as follows:

$d_L^{rnd} \longleftarrow r + d_L^{rnd}$, and

$d_{L-1} \longleftarrow d_{L-1} - 1$; and

otherwise, if $d_L^{rnd} \geq r$, then modifying d_L^{rnd} and d_{L-1} as follows:

$d_L^{rnd} \longleftarrow d_L^{rnd} - r$, and

$d_{L-1} \longleftarrow d_{L-1} + 1$.

8. The method of claim 1, wherein:

the digits are restricted to the digit set {*-r+1, -r+2, ..., r-2*}.

9. A method comprising:

receiving one or more operands upon which an operation is to be performed using a digit-recurrence procedure employing on-the-fly rounding;

providing a first storage location in which an intermediate result, R_j, of the operation is maintained as a sum of intermediate digits, $d_1 r^{-1} + d_2 r^{-2} + \ldots + d_j r^{-j}$, where

r represents a radix associated with the digit-recurrence procedure, j represents a current iteration of the digit-recurrence procedure, and d_j represents a digit generated during iteration j of the digit-recurrence procedure;

providing a second storage location in which a value, R_j^-, is maintained representing $R_j - r^{-j}$;

during iterations *1* to *L-2* of the digit-recurrence procedure, where L represents a number of digits that satisfies a predetermined precision for the operation, (1) generating d_j, and (2) updating R_j and R_j^- based upon d_j, r, and R_{j-1} and R_{j-1}^-; and

determining a rounded fractional result based upon (1) digits d_{L-1} and d_L and (2) R_{L-2} or R_{L-2}^-.

10. The method of claim 9, wherein the operation comprises division.

11. The method of claim 9, wherein the operation comprises square-root.

12. The method of claim 9, wherein a value, R_j^+, representing $R_j + r^{-j}$ is not maintained during iterations *1* to *L-2* of the digit-recurrence procedure, thereby saving storage typically allocated to R_j^+ in conventional digit-recurrence procedures employing on-the-fly rounding.

13. The method of claim 9, wherein the first and second storage locations comprise shift registers, and wherein said updating R_j and R_j^- based upon d_j, r, and R_{j-1} and R_{j-1}^- comprises shifting the contents of the shift registers and appending new digits.

14. The method of claim 9, wherein:

$$R_j = \begin{cases} (R_{j-1}, d_j) & d_j \geq 0 \\ (R_{j-1}^-, d_j + r) & d_j \leq -1 \end{cases}; \text{ and}$$

$$R_j^- = \begin{cases} (R_{j-1}, d_j - 1) & d_j \geq 1 \\ (R_{j-1}^-, d_j + r - 1) & d_j \leq 0 \end{cases}.$$

15. The method of claim 9, wherein said determining a rounded fractional result

based upon (1) digits d_{L-1} and d_L and (2) R_{L-2} or R_{L-2}^- further comprises:

generating the digits d_{L-1} and d_L;

generating a rounded last digit d_L^{rnd};

if $d_L^{rnd} \leq -1$, then modifying d_L^{rnd} and d_{L-1} as follows:

$d_L^{rnd} \longleftarrow r + d_L^{rnd}$, and

$d_{L-1} \longleftarrow d_{L-1} - 1$; and

otherwise, if $d_L^{rnd} \geq r$, then modifying d_L^{rnd} and d_{L-1} as follows:

$d_L^{rnd} \longleftarrow d_L^{rnd} - r$, and

$d_{L-1} \longleftarrow d_{L-1} + 1$.

16. The method of claim 9, wherein:

the digits are constrained to the digit set {*-r+1, -r+2, ..., r-2*}.

17. A method comprising the steps of:

receiving one or more operands upon which an operation is to be performed using a digit-recurrence procedure with on-the-fly rounding;

providing a first storage location in which an intermediate result of the operation, R_j, is maintained, where,

$R_j = d_1 r^{-1} + d_2 r^{-2} + \ldots + d_j r^{-j}$ converges to an infinitely precise result as j tends to infinity,

r represents a radix associated with the digit-recurrence procedure,

j represents a current iteration of the digit-recurrence procedure, and

d_j represents a digit generated during iteration j of the digit-recurrence procedure;

providing a second storage location in which a value, R_j^-, is maintained representing $R_j - r^{-j}$;

an initialization step for constructing R_0 and R_0^-;

a main iteration step for generating digits d_1 through d_j and updating R_j and R_j^-

during iterations *1* to *L-2* of the iterative digit-recurrence procedure, where *L* represents a number of digits that satisfies a predetermined precision for the operation; and

a step for determining a rounded fractional result based upon (1) digits d_{L-1} and d_L and (2) R_{L-2} or R_{L-2}^-.

18. The method of claim 17, wherein:

$$R_j = \begin{cases} (R_{j-1}, d_j) & d_j \geq 0 \\ (R_{j-1}^-, d_j + r) & d_j \leq -1 \end{cases}; \text{ and}$$

$$R_j^- = \begin{cases} (R_{j-1}, d_j - 1) & d_j \geq 1 \\ (R_{j-1}^-, d_j + r - 1) & d_j \leq 0 \end{cases}.$$

19 ~~18~~. The method of claim 17, wherein said step for determining a rounded fractional result based upon (1) digits d_{L-1} and d_L and (2) R_{L-2} or R_{L-2}^- further comprises:

generating the digits d_{L-1} and d_L;

generating a rounded last digit d_L^{rnd};

if $d_L^{rnd} \leq -1$, then modifying d_L^{rnd} and d_{L-1} as follows:

$d_L^{rnd} \longleftarrow r + d_L^{rnd}$, and

$d_{L-1} \longleftarrow d_{L-1} - 1$; and

otherwise, if $d_L^{rnd} \geq r$, then modifying d_L^{rnd} and d_{L-1} as follows:

$d_L^{rnd} \longleftarrow d_L^{rnd} - r$, and

$d_{L-1} \longleftarrow d_{L-1} + 1$.

20 ~~19~~. An apparatus comprising:

a first storage means for storing an intermediate result, R_j, of an operation on one or more operands, the operation implemented as a digit recurrence procedure with on-the-fly rounding, where $R_j = d_1 r^{-1} + d_2 r^{-2} + \ldots + d_j r^{-j}$, *r* represents a radix associated with the digit-recurrence procedure, *j* represents a current iteration of the digit-recurrence procedure, and d_j represents a digit generated during iteration *j* of the digit-recurrence procedure;

a second storage means, coupled to the first storage means, for storing a value, R_j^-, representing $R_j - r^{-j}$;

an update means, coupled to the first and second storage means, for updating R_j and R_j^- based upon d_j, *r*, and R_{j-1} and R_{j-1}^- during iterations *1* to *L-2* of the digit-recurrence procedure;

a digit selection means, coupled to the update means, for generating d_j during iterations *1* to *L-2* of the digit-recurrence procedure, where *L* represents a number of digits that satisfies a predetermined precision for the operation; and

means, coupled to the first and second storage means, for determining a rounded fractional result based upon (1) digits d_{L-1} and d_L and (2) R_{L-2} or R_{L-2}^-.

~~20~~. The apparatus of claim 19, wherein the operation comprises a division operation or a square-root operation.

~~21~~. The apparatus of claim 19, wherein the digit selection means supports both division and square-root operations.

~~22~~. The digit recurrence unit of claim 19, further comprising a delay element coupled to the output of the digit selection means to hold at least two iterations of digits.

23. An apparatus comprising:

a first register to store an intermediate result, R_j, of an operation on one or more operands, the operation implemented as a digit recurrence procedure with on-the-fly rounding, where $R_j = d_1 r^{-1} + d_2 r^{-2} + \ldots + d_j r^{-j}$, r represents a radix associated with the digit-recurrence procedure, j represents a current iteration of the digit-recurrence procedure, and d_j represents a digit generated during iteration j of the digit-recurrence procedure;

a second register, coupled to the first register, to store a value, R_j^-, representing $R_j - r^{-j}$;

a digit selection lookup table to generating d_j during iterations *1* to *L-2* of the digit-recurrence procedure, where *L* represents a number of digits that satisfies a predetermined precision for the operation;

load and shift control logic and on-the-fly conversion logic to update R_j and R_j^- based upon d_j, r, and R_{j-1} and R_{j-1}^- during iterations *1* to *L-2* of the digit-recurrence procedure; and

final rounding logic to determine a rounded fractional result based upon (1) digits d_{L-1} and d_L and (2) R_{L-2} or R_{L-2}^-.

24. The apparatus of claim 23, wherein the operation comprises a division operation or a square-root operation.

25. The apparatus of claim 23, wherein the digit selection lookup table supports both division and square-root operations.

26. The apparatus of claim 23, further comprising a delay element interposed between the digit selection lookup table and the load and shift control logic and on-the-fly conversion logic to hold at least two iterations of digits.

~~27~~ 28. A machine-readable medium having stored thereon data representing sequences of instructions, the sequences of instructions which, if executed by a processor, cause the processor to:

maintain a first form of an intermediate result of an operation in a first register;

maintain a second form of the intermediate result in a second register;

update each of the first form and the second form of the intermediate result by register swapping or concatenation responsive to receiving digits *1* to *L-2* of the intermediate result from a digit recurrence unit, where *L* represents a number of digits that satisfies a predetermined precision for the operation; and

generate a rounded result by determining digits d_{L-1} and d_L and appending a rounded last digit to either the first form of the intermediate result or the second form of the intermediate result.

~~28~~ 29. The machine-readable medium of claim 27, wherein the first form of the intermediate result comprises a value R_j representing a sum of intermediate digits, $d_1 r^{-1} + d_2 r^{-2} + ... + d_j r^{-j}$, which converges to an infinitely precise result as *j* tends to infinity, where *r* represents a radix associated with a digit-recurrence algorithm implemented by the digit-recurrence unit, *j* represents a current iteration of the digit recurrence algorithm, and d_j represents a digit generated during iteration *j* of the digit-recurrence algorithm.

~~29~~ 30. The machine-readable medium of claim 28, wherein the second form of the intermediate result comprises a value R_j^-, where R_j^- differs from R_j by a unit of the j^{th} position, r^{-j}.